

Optimization of Memory Usage in Tardos's Fingerprinting Codes

Koji Nuida, Manabu Hagiwara, Hajime Watanabe, and Hideki Imai, *Fellow, IEEE*

Abstract—It is known that Tardos's collusion-secure probabilistic fingerprinting code (Tardos code; STOC'03) has length of theoretically minimal order with respect to the number of colluding users. However, Tardos code uses certain continuous probability distribution in codeword generation, which creates some problems for practical use, in particular, it requires large extra memory. A solution proposed so far is to use some finite probability distributions instead. In this paper, we determine the optimal finite distribution in order to decrease extra memory amount. By our result, the extra memory is reduced to $1/32$ of the original, or even becomes needless, in some practical setting. Moreover, the code length is also reduced, e.g. to about 20.6% of Tardos code asymptotically. Finally, we address some other practical issues such as approximation errors which are inevitable in any real implementation.

Index Terms—Collusion-secure code, Tardos code, memory optimization, digital rights managements

I. INTRODUCTION

RECENT progress in information technology has enabled us to handle easily commercial objects (such as movies, musics, customers' data) in a digital form. This increased our convenience dramatically, however as the amount of such digital contents constantly grows, information leakage and counterfeiting, in particular those caused by authorized users, have become a serious concern. Prevention of such illegal copying is often difficult by either technological or social reason. An alternative solution is to embed user identification information into each content by watermarking technique, making the guilty user (called a "pirate") traceable from the leaked content without decreasing convenience for innocent users too much. For this purpose, it was pointed out ([1]) that the embedded information should be designed securely against "collusion-attacks", that is a kind of modification of embedded information by a group of pirates. A *c-secure code* provides such identification information which is secure against *c* pirates or less.

It is known that Tardos's probabilistic *c*-secure code [12] (Tardos code) has length of theoretically minimal order among all possible *c*-secure codes with respect to *c*. The frequency of 0s and 1s in the codewords is decided by outputs of

certain probability distribution, which is referred to as the *bias distribution* in this paper. Tardos's work is a milestone in this research area because of the theoretical impact, however there are some hurdles for practical implementation, due to the property that Tardos's bias distributions are continuous. An explicit implementation of continuous distributions would be impossible, while effects of approximation of bias distributions on the security performance have not yet been evaluated. Moreover, (approximated values of) the outputs of the bias distribution, which should be of high accuracy to make the code *c*-secure, are supposed to be recorded throughout. Thus large amount of extra memory is required for a practical use.

A simple solution is to replace the continuous bias distributions with *finite* ones. For instance, a bias distribution with 4 possible outputs needs only 2 bits of memory to record one output, i.e. to record "which of the four". This solution was first explored by Hagiwara, Hanaoka and Imai in [6]; they established a formula of sufficient code length in terms of a given (finite) bias distribution and desired security performance. They also proposed a "*c*-indistinguishability" condition for suitable bias distributions, with three concrete examples that reduce the code lengths to about 60% of Tardos codes. However, it has not yet been discussed whether their choice of bias distributions is optimal for the purpose of reducing extra memory amount. Moreover, a problem concerning practical implementation is left unsolved as well: their code requires calculation of some "score" of each user, which cannot be explicitly representable in general by usual number systems on computers (e.g. floating-point numbers), however effects of approximation of scores have not been evaluated so far.

The aim of this paper is to solve the abovementioned problems. First, we exhibit a strong evidence that the code lengths decrease substantially due to *c*-indistinguishability condition. Thus, we may restrict our attention to bias distributions satisfying this condition. Secondly, we determine the set of *c*-indistinguishable bias distributions, together with the set of the optimal ones among them (namely, those with minimal number of possible outputs). We show that the optimal distribution has only $\lceil c/2 \rceil$ possible outputs, where $\lceil x \rceil$ denotes as usual the smallest integer n with $x \leq n$; thus only $\lceil \log_2 \lceil c/2 \rceil \rceil$ -bits memory are required to record one output. (Table I gives a numerical example, where bias distributions for Tardos codes are approximated by single-precision binary floating-point numbers.) This shows that our result reduces the extra memory amount significantly; in particular, it even makes such extra memory needless when $c = 2$. Moreover, we improve the code length formula in [6] to reduce code lengths further and to evaluate effects of approximation of users' scores. The

K. Nuida, M. Hagiwara and H. Watanabe are with Research Center for Information Security (RCIS), National Institute of Advanced Industrial Science and Technology (AIST), Japan.

H. Imai is also with AIST-RCIS, and is with Faculty of Scientific Engineering, Chuo University, Japan.

This paper is the full version of the authors' work presented in: 9th Information Hiding, Saint-Malo, France, June 13, 2007.

This study has been sponsored by the Ministry of Economy, Trade and Industry, Japan (METI) under contract, New-generation Information Security R&D Program, and by JSPS Grants-in-Aid for Scientific Research.

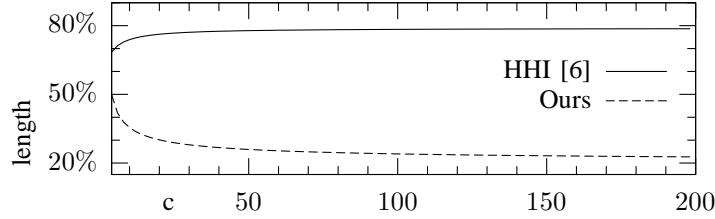


Fig. 1. Ratio of code lengths relative to Tardos codes

combination of our new formula and our optimal distributions provides much shorter code lengths than Tardos codes and than [6] (see Figure 1). We also investigate the asymptotic behavior of our code length; the ratio of our code length relative to Tardos code converges to about 20.6% as $c \rightarrow \infty$.

TABLE I
A COMPARISON OF REQUIRED EXTRA MEMORY AMOUNT

Case 1: 2 pirates, 200 users, error probability $\leq 10^{-11}$.
Case 2: 4 pirates, 400 users, error probability $\leq 10^{-11}$.

		bits / position	code length	total bits
Case 1	Tardos	32	12 400	396 800
	Ours	0	6278	0
	%	0	50.6	0
Case 2	Tardos	32	51 200	1 638 400
	Ours	1	19 750	19 750
	%	3.1	38.6	1.2

This paper is organized as follows. After some preliminary (Section II) on the model of c -secure codes, some preceding works, problems and notations, we observe in Section III the importance of the c -indistinguishability condition. Section IV-A shows some properties of c -indistinguishable distributions; Section IV-B determines the set of c -indistinguishable distributions; and Section IV-C determines the set of the optimal distributions. Section V-A gives our improvement of the code length formula established in [6]; Section V-B investigates the asymptotic behavior of our code length; and Section V-C provides some numerical examples. We give remarks on some recent related works in Section VI. Finally, Appendices are given and devoted to the proofs of some of our results.

II. PRELIMINARIES

A. Our Model for Collusion-Secure Codes

In this subsection, we describe our model for collusion-secure codes. In our model, a content server embeds a binary codeword $w_i = (w_{i,1}, \dots, w_{i,m})$ of length m into a digital content, which will be distributed to i -th user u_i , by certain watermarking technique. *Pirates*, who are the adversarial users attacking the code, then make an illegal copy of the distributed content which involves a codeword possibly modified by them. When the illegally copied content is found, the content server first extracts the embedded codeword $y = (y_1, \dots, y_m)$ (called the *pirated codeword*). Some bits y_j in y may be broken and hence not decodable; such a bit is denoted by ‘?’. Then the server executes a *tracing algorithm* for detecting the pirates, with the y and all the w_i s as input.

Regarding the attack model, we assume that ℓ pirates try to detect the positions of (parts of) the embedded codeword from differences of their contents, and then to modify bits of the codeword in these positions by some (possibly probabilistic) algorithm, called a *pirates’ strategy*. This attack model is formulated as the following assumption, which was originally introduced in [1] and has been adopted in most of the preceding works (e.g. [1], [6], [12]):

Assumption 1 (Marking Assumption): If all the bits $w_{i_1,j}, \dots, w_{i_\ell,j}$ in codewords of the pirates $u_{i_1}, \dots, u_{i_\ell}$ at the same, say j -th position coincide (we call such a position *undetectable*), then $y_j = w_{i_1,j}$.

Moreover, we also put the following assumption:

Assumption 2 (Pirates’ Knowledge): Pirates have no information on the actual choice of innocent (i.e. non-pirate) users’ codewords, other than their *a priori* distribution which may be publicly known. As a result, the choice of y is independent of those codewords.

To discuss the security performance of our codes, we fix the meaning of the following terms: *false-negative* means that the tracing algorithm outputted no pirates; *false-positive* means that the tracing algorithm outputted at least one innocent user; *tracing error* means that false-negative or false-positive (or possibly both) occurs. A code equipped with a tracing algorithm is called *c-secure (with ε -error)* if the tracing error probability is bounded by a negligibly small value ε provided the number of pirates is at most c .

B. Tardos Code and Its Generalization

In this subsection we summarize the code construction and tracing algorithms of c -secure Tardos code [12] and its generalization given in [6] as follows. First, the content server is supposed to choose the random values $0 < p^{(j)} < 1$ independently for every $1 \leq j \leq m$, according to a given probability distribution \mathcal{P} which we refer to as the *bias distribution*. (Details for the choices of \mathcal{P} in these codes are irrelevant here and hence omitted; see the original papers for details.) Here we only treat the bias distributions whose output values are in the open interval $(0, 1)$ and which are *symmetric* in the following sense; we have

$$\text{Prob}((\mathcal{P}\text{'s output}) \in A) = \text{Prob}((\mathcal{P}\text{'s output}) \in 1 - A)$$

for any subset A of the interval $(0, 1)$, where *Prob* signifies the probability and $1 - A = \{1 - a \mid a \in A\}$. (When \mathcal{P} is finite, it is symmetric in this sense if and only if it outputs a and $1 - a$ with the same probability for any a .) The resulting sequence $P = (p^{(1)}, \dots, p^{(m)})$ should be stored and be kept secret throughout

the scheme: due to Assumption 2, pirates may be allowed to guess the values $p^{(j)}$ from public information on \mathcal{P} and the pirates' codewords, but not to know about the actual choices of $p^{(j)}$. Then, secondly, the server chooses each codeword w_i in the following probabilistic manner: $\text{Prob}(w_{i,j} = 1) = p^{(j)}$ and $\text{Prob}(w_{i,j} = 0) = 1 - p^{(j)}$ for j -th position. All the bits $w_{i,j}$ are supposed to be independently chosen.

In the tracing algorithm, the server calculates a score S_i of each user u_i by $S_i = \sum_{j=1}^m S_i^{(j)}$, where

$$S_i^{(j)} = \begin{cases} \sigma(p^{(j)}) & \text{if } (y_j, w_{i,j}) = (1, 1) , \\ -\sigma(1 - p^{(j)}) & \text{if } (y_j, w_{i,j}) = (1, 0) , \\ 0 & \text{if } y_j \in \{0, ?\} , \end{cases}$$

with $\sigma(p) = \sqrt{(1-p)/p}$ for $0 < p < 1$. The output of the tracing algorithm is then the (possibly empty) list of *all* users u_i with $S_i \geq Z$, where Z is a suitably selected threshold parameter. Details of the choices of Z are also omitted here.

C. Problems

A problem of Tardos code is, as we mentioned in the Introduction, that the bias distribution \mathcal{P} used in his codeword generation is *continuous*. An explicit implementation of such a \mathcal{P} seems to be impossible. Moreover, even if we would like to approximate this \mathcal{P} , e.g. by floating-point numbers, the original security proof does not concern effects of such inevitable approximation; and large amount of extra memory is required to record the approximated values of \mathcal{P} since these values should be of high accuracy to make the code c -secure.

A solution proposed in [6] is to use suitable *finite* bias distributions instead. They gave formulae of code length and threshold parameter corresponding to a given finite bias distribution \mathcal{P} , making the code c -secure. Moreover, by observing the form of their formula, they also proposed a “ c -indistinguishability” condition for \mathcal{P} which would be effective to reduce the code lengths.

However, regarding abovementioned memory problem, the following question remained open: Is their choice of \mathcal{P} optimal in terms of required memory? Moreover, although the users' scores are irrational numbers in general, effects of approximation of scores on the tracing error probability has not yet been discussed. In the rest of this paper, we give solutions for these problems.

D. Notations

This subsection summarizes some notations used throughout this paper. First, let the following expression

$$\{(val, prob) \mid cond\}$$

signify the probability distribution such that a value val is taken with probability $prob$, where val and $prob$ vary subject to the condition $cond$. Given a finite bias distribution \mathcal{P} , let p_0, p_1, \dots, p_k denote the possible outputs in increasing order and write $q_i = \text{Prob}(\mathcal{P} \text{ outputs } p_i)$; thus $p_{k-i} = 1 - p_i$ and

$q_{k-i} = q_i$ by the symmetry of \mathcal{P} . For $1 \leq \ell \leq c$ and $0 \leq x \leq \ell$, define functions $f_{\ell,x}(p)$ and $g_{\ell,x}(p)$ for $0 < p < 1$ by

$$\begin{aligned} f_{\ell,x}(p) &= p^x (1-p)^{\ell-x} (x\sigma(p) - (\ell-x)\sigma(1-p)) , \\ g_{\ell,x}(p) &= xp^{x-1} (1-p)^{\ell-x} - (\ell-x)p^x (1-p)^{\ell-x-1} , \end{aligned}$$

where

$$\sigma(p) = \sqrt{(1-p)/p} .$$

These two functions satisfy the following relation $f_{\ell,x}(p) = g_{\ell,x}(p)\sqrt{p(1-p)}$. Put

$$R_{\ell,x} = \max\{0, E_p[f_{\ell,x}(p)]\} ,$$

where E_p signifies the expected value over outputs p of \mathcal{P} . Define a function $r(t)$ by

$$r(t) = (e^t - 1 - t)/t^2 \text{ for } t > 0 ,$$

and let

$$\mathcal{R}_{\ell,\mathcal{P}} = E_p[-f_{\ell,0}(p)] - \sum_{x=1}^{\ell-1} \binom{\ell}{x} R_{\ell,x} \text{ for } 1 \leq \ell \leq c .$$

Moreover, $\log = \log_e$ denotes the natural logarithm, and $\lceil x \rceil$ denotes the smallest integer n such that $n \geq x$.

III. A CHARACTERIZATION OF THE c -INDISTINGUISHABILITY

Before solving the problems mentioned in Section II-C, we investigate the c -indistinguishability condition for bias distributions proposed in [6]. This condition was introduced for the purpose of reducing code lengths determined by the formula given in [6]; however, it has not yet been discovered how much this condition contributes to decreasing the *true* tracing error probability (and hence to reducing the code length). This section exhibits a strong evidence that this condition is in fact substantial for decreasing the error probability.

First, we recall from [6] the following definition of the c -indistinguishability condition (see Section II-D for notations):

Definition 1: A (finite) bias distribution \mathcal{P} is called c -indistinguishable, or c -ind in short, if $\sum_{x=1}^{\ell-1} \binom{\ell}{x} R_{\ell,x} = 0$ for all $2 \leq \ell \leq c$.

Remark 1: Since the value $R_{\ell,x}$ is always nonnegative by definition, this condition is equivalent to $R_{\ell,x} = 0$ (or equivalently, $E_p[f_{\ell,x}(p)] \leq 0$) for all $2 \leq \ell \leq c$ and $1 \leq x \leq \ell - 1$.

Then we show that all attack strategies have the same efficiency on average *if and only if* \mathcal{P} is c -ind. This claim implies a substantial significance of the c -ind condition.

We start with an arbitrary finite bias distribution \mathcal{P} . Let u_1, \dots, u_ℓ (where $\ell \leq c$) be the pirates and w_1, \dots, w_ℓ their codewords. The pirates would hope none of them being outputted by the tracing algorithm, therefore they would try to create the pirated codeword y so that all of their scores will be as small as possible. For this purpose, it is necessary for the sum S of their scores to be small. By the definition of the tracing algorithm, S can be decomposed as $S = S' + S''$, where S' denotes the sum of pirates' bitwise scores over the undetectable positions, which is independent of the pirates'

strategy due to Marking Assumption (Assumption 1), and S'' is the sum over the remaining positions j with $y_j = 1$.

Now for $1 \leq j \leq m$ and $I \subset \{1, 2, \dots, \ell\}$, let \mathcal{B}_I denote the event that $w_{i,j} = 1$ for $i \in I$ and $w_{i,j} = 0$ for $i \notin I$, and let \mathcal{B}'_I be the event that \mathcal{B}_I occurs and $y_j = 1$. Then the contribution of j -th position of their codewords for S'' under the event \mathcal{B}'_I (where $I \neq \emptyset$ and $I \neq \{1, \dots, \ell\}$) is $x\sigma(p^{(j)}) - (\ell - x)\sigma(1 - p^{(j)})$, where $x = |I|$. Thus its expected value conditioned on \mathcal{B}'_I over the choices of $p^{(j)}$ is given by

$$\sum_p \text{Prob}(p^{(j)} = p \mid \mathcal{B}'_I) (x\sigma(p) - (\ell - x)\sigma(1 - p)) \quad (1)$$

where the sum is taken over all possible outputs p of \mathcal{P} . Under this setting, our claim is expressed as the following proposition, whose proof is postponed until the end of Section IV-A since it requires some results given in that section:

Proposition 1: The expected value (1) is always 0 if and only if \mathcal{P} is c -indistinguishable.

Based on this observation, we restrict our attention to c -ind bias distributions from now on.

IV. THE OPTIMAL BIAS DISTRIBUTION

A. Properties of the c -Indistinguishability Condition

In this subsection, we investigate properties of the c -indistinguishability condition as a preliminary for the following sections. Proofs will be given in Appendix A.

Let \mathcal{P} be a (finite) bias distribution. First, a straightforward observation can show that $g_{\ell, \ell-x}(p) = -g_{\ell, x}(1 - p)$ and $f_{\ell, \ell-x}(p) = -f_{\ell, x}(1 - p)$, therefore by symmetry of \mathcal{P} we have

$$E_p[f_{\ell, \ell-x}(p)] = -E_p[f_{\ell, x}(1 - p)] = -E_p[f_{\ell, x}(p)] \quad (2)$$

This infers the following result concerning the case when a bias distribution becomes c -ind:

Proposition 2: Let \mathcal{P} be a (finite) bias distribution.

1) \mathcal{P} is c -ind if and only if

$$E_p[f_{\ell, x}(p)] = 0 \quad (3)$$

for any $2 \leq \ell \leq c$ and $1 \leq x \leq \ell - 1$.

2) If ℓ is even, then (3) always holds for $x = \ell/2$. In particular, \mathcal{P} is always 2-ind (cf. [6, Proposition 1]).

3) Condition (3) holds for an ℓ and $x = x_0$ if and only if (3) holds for this ℓ and $x = \ell - x_0$.

The following recursive relations for $f_{\ell, x}$ and $g_{\ell, x}$ are key ingredients of our argument in this section:

Lemma 1: We have $f_{\ell-1, x}(p) = f_{\ell, x}(p) + f_{\ell, x+1}(p)$ and $g_{\ell-1, x}(p) = g_{\ell, x}(p) + g_{\ell, x+1}(p)$ for $0 \leq x \leq \ell - 1$.

From this lemma, we derive the following properties. First, the next proposition says that the c -ind condition simplify the value $\mathcal{R}_{\ell, \mathcal{P}}$ and makes it positive:

Proposition 3: If \mathcal{P} is a c -ind distribution, then we have $\mathcal{R}_{\ell, \mathcal{P}} = E_p[\sqrt{p(1-p)}] > 0$ for $1 \leq \ell \leq c$.

Secondly, the next lemma reduces the complexity to determine whether a given bias distribution is c -ind:

Lemma 2: If the condition (3) is satisfied for any two of the three pairs of parameters $(\ell, x) = (\ell' - 1, x')$, (ℓ', x')

and $(\ell', x' + 1)$, then this condition is also satisfied for the remaining one.

Now we are able to prove the following result, which can be seen as a generalization of [6, Proposition 1] since any bias distribution is 1-ind by definition:

Proposition 4: If c is odd, then any c -ind bias distribution is also $(c + 1)$ -ind.

Moreover, the following criterion of the c -ind condition is deduced from the above results:

Proposition 5: Let $c \geq 3$, and let c' denote the largest odd number such that $c' \leq c$.

- 1) If (3) is satisfied for all parameters of the form $(\ell, x) = (c', x)$ with $1 \leq x \leq (c' - 1)/2$, then \mathcal{P} is c -ind.
- 2) If \mathcal{P} is $(c' - 2)$ -ind and (3) is satisfied for at least one parameter of the form (c', x_0) with $1 \leq x_0 \leq c' - 1$, then \mathcal{P} is also c -ind. In particular, \mathcal{P} is c -ind if for each odd number ℓ with $3 \leq \ell \leq c'$, the condition (3) is satisfied for at least one parameter of the form (ℓ, x_ℓ) .

At the end of this subsection, we give the postponed proof of Proposition 1 in Section III:

Proof of Proposition 1: First, since the value $p^{(j)}$ is assumed to be secret for the pirates, the conditional probability $\text{Prob}(y_j = 1 \mid \mathcal{B}_I \wedge (p^{(j)} = p))$ is constant on outputs p of \mathcal{P} , which is equal to $\text{Prob}(y_j = 1 \mid \mathcal{B}_I)$. On the other hand, we have

$$\text{Prob}(\mathcal{B}_I \mid p^{(j)} = p) = p^x(1 - p)^{\ell - x}$$

by the codeword generation. Thus by putting

$$C = \text{Prob}(y_j = 1 \mid \mathcal{B}_I) / \text{Prob}(\mathcal{B}'_I) \quad ,$$

we have

$$\begin{aligned} \text{Prob}(p^{(j)} = p \mid \mathcal{B}'_I) &= C \cdot \text{Prob}((p^{(j)} = p) \wedge \mathcal{B}_I) \\ &= C \cdot \text{Prob}(p^{(j)} = p) p^x (1 - p)^{\ell - x} \end{aligned}$$

since $\mathcal{B}'_I = (\mathcal{B}_I \wedge (y_j = 1))$, therefore (1) is equal to

$$C \sum_p \text{Prob}(p^{(j)} = p) f_{\ell, x}(p) = C \cdot E_p[f_{\ell, x}(p)] \quad .$$

Thus by Proposition 2, this value is always 0 regardlessly of the pirates' strategy if and only if \mathcal{P} is c -ind. ■

B. Determining c -Indistinguishable Distributions

In this subsection, we determine all the c -ind bias distributions \mathcal{P} for every c , by proving in Theorem 1 below that the c -ind bias distributions are in one-to-one correspondence with objects defined as follows:

Definition 2: We refer to a pair $\mathcal{Q} = (X, \omega)$ of a finite subset X of the open interval $(-1, 1)$ and a positive function $\omega > 0$ on X as a *quadrature system*, or a *QS* in short, of degree d if we have

$$\int_{-1}^1 F(t) dt = \sum_{\xi \in X} \omega(\xi) F(\xi) \quad (4)$$

for any real polynomial $F(t)$ of degree less than or equal to d . We refer to the size $|X|$ of X as the *order* of \mathcal{Q} , and we say

that \mathcal{Q} is *symmetric* if $-X = X$ (where $-X = \{-\xi \mid \xi \in X\}$) and $\omega(-\xi) = \omega(\xi)$ for all $\xi \in X$.

Example 1: Let $X = \{0, \pm\sqrt{15}/5\}$, $\omega(0) = 8/9$ and $\omega(\pm\sqrt{15}/5) = 5/9$. Then a direct calculation can verify that (4) holds for any $F(t)$ of degree less than or equal to 5. Thus (X, ω) is a symmetric QS of order 3 and degree 5 in the sense of Definition 2.

Now we give the aforementioned theorem on the one-to-one correspondence as follows, which will be proved in Appendix B:

Theorem 1: For each c , the c -ind bias distributions are in one-to-one correspondence with the symmetric QSs of degree $c - 1$. More precisely:

- For a symmetric QS $\mathcal{Q} = (X, \omega)$ of degree $c - 1$, define a probability distribution $\mathcal{P}(\mathcal{Q})$ by

$$\mathcal{P}(\mathcal{Q}) = \left\{ \left(\frac{1+\xi}{2}, \frac{\omega(\xi)}{C\sqrt{1-\xi^2}} \right) \mid \xi \in X \right\}$$

(see Section II-D for notation), where we put $C = \sum_{\xi \in X} \omega(\xi) / \sqrt{1-\xi^2}$.

- For a c -ind bias distribution $\mathcal{P} = \{(p_i, q_i) \mid 1 \leq i \leq k\}$, define a pair $\mathcal{Q}(\mathcal{P}) = (\{\xi_1, \dots, \xi_k\}, \omega)$ by putting, for $1 \leq i \leq k$,

$$\xi_i = 2p_i - 1 \text{ and } \omega(\xi_i) = \frac{\sqrt{p_i(1-p_i)}q_i}{C'},$$

where we put $C' = \sum_{i=1}^k \sqrt{p_i(1-p_i)}q_i/2$.

Then $\mathcal{P}(\mathcal{Q})$ is c -ind, $\mathcal{Q}(\mathcal{P})$ is a symmetric QS of degree $c - 1$, $\mathcal{Q}(\mathcal{P}(\mathcal{Q})) = \mathcal{Q}$ and $\mathcal{P}(\mathcal{Q}(\mathcal{P})) = \mathcal{P}$.

Remark 2: Note that any symmetric QS of even degree $2d$ is also a QS of degree $2d + 1$ by the definition of QSs. This fact corresponds to Proposition 4 via the one-to-one correspondence in Theorem 1.

C. The Optimal c -Indistinguishable Distribution

Among the c -ind bias distributions, in this subsection we determine the optimal ones for the purpose of reducing extra memory amount. Owing to Proposition 4, we may concentrate our attention on the case when c is even.

First, as we mentioned in the Introduction, the optimal c -ind distributions are precisely the ones with minimal number of possible outputs. By Theorem 1, such c -ind distributions correspond to the symmetric QSs of degree $c - 1$ with minimal order; thus our task here is to determine those QSs. However, in fact the solution of this problem has been given (in different terminology) as the following classical result:

Theorem 2 (e.g. [3], [11]): For $\nu \geq 1$, let

$$L_\nu(t) = \frac{1}{2^\nu \nu!} \left(\frac{d}{dt} \right)^\nu (t^2 - 1)^\nu$$

be the ν -th Legendre polynomial normalized as $L_\nu(1) = 1$. Let X be the set of zeroes of $L_\nu(t)$ (i.e. values x with $L_\nu(x) = 0$), and put

$$\omega(\xi) = \frac{2}{(1-\xi^2)L_\nu'(\xi)^2} \text{ for } \xi \in X$$

(see [3, Section 7.3.1, p.316] for the expression of $\omega(\xi)$). Then $\mathcal{Q}_\nu = (X, \omega)$ is the unique symmetric QS of minimal order

subject to the degree being $2\nu - 1$; namely, it is a symmetric QS of order ν and degree $2\nu - 1$, while no other QS of degree $2\nu - 1$ has order less than or equal to ν .

For instance, \mathcal{Q}_3 is the QS shown in Example 1. We refer to the QS \mathcal{Q}_ν defined in this theorem as the *Gauss-Legendre QS*, or the *GL QS* in short, because of its deep relationship to the ‘‘Gauss-Legendre quadrature formula’’, that is a classical approximation method for integral (see e.g. [11]). Now by combining Theorems 1 and 2, we determine the optimal bias distribution (which we refer to as the *Gauss-Legendre distribution*, or the *GL distribution* in short) explicitly as follows:

Theorem 3: For $\nu \geq 1$, let

$$\tilde{L}_\nu(t) = \left(\frac{d}{du} \right)^\nu (u^2 - 1)^\nu \Big|_{u=2t-1},$$

a polynomial in t of degree ν . Then the unique optimal (2ν) -ind distribution $\mathcal{P} = \mathcal{P}_{2\nu}$ is given by

$$\left\{ \left(p, \frac{C}{(p(1-p))^{3/2} \tilde{L}_\nu'(p)^2} \right) \mid \tilde{L}_\nu(p) = 0 \right\}$$

(see Section II-D for notation), where C is the normalizing constant adjusting the total probability to 1. This $\mathcal{P}_{2\nu}$ has ν possible outputs.

The proof of this theorem will be given in Appendix C. Table II shows the explicit GL distributions for small c , where the output values less than $1/2$ are omitted by symmetry.

Remark 3: By Theorem 3, the optimal c -ind distribution \mathcal{P}_c has $\lceil c/2 \rceil$ possible outputs, therefore only $\lceil \log_2 \lceil c/2 \rceil \rceil$ -bits of memory are sufficient to record one value $p^{(j)}$ (whenever a relatively small table of possible outputs of \mathcal{P}_c is held together). As we mentioned in the Introduction, some comparison of the required memory amount between Tardos code and ours is shown in Table I above, where we put $c = 2$, $N = 200$ and $\varepsilon = 10^{-11}$ in Case 1, and $c = 4$, $N = 400$ and $\varepsilon = 10^{-11}$ in Case 2. Here the code lengths of our codes are calculated by using a formula given in the next section; and we assume that outputs of Tardos’s continuous bias distributions are approximated by using single-precision (4-bytes) floating-point numbers. The table shows that our optimal bias distributions in fact reduce the memory amount dramatically. Note that our optimal distributions require 4-bytes of memory or more to record one output in the (very impractical) case $c \geq 2^{32} + 1 = 4\,294\,967\,297$. However, for such c , the approximation of Tardos’s distributions require much larger memory in order to attain comparable security.

V. CODE LENGTHS

A. An Improved Formula for Code Lengths

In this subsection, we improve the formula for code lengths and thresholds given in [6] to reduce code lengths. Also, we slightly modify the tracing algorithm to evaluate the effects of approximation of users’ scores. Here we do *not* assume that the bias distribution \mathcal{P} is the optimal one determined in the previous section, since it is generally inevitable in practical implementation to perform some approximation of the optimal bias distribution (cf. Table II).

TABLE II
THE OPTIMAL c -IND DISTRIBUTIONS \mathcal{P}_c

c	$\tilde{L}(t)$	value	probability
2	$2(2t-1)$	$1/2$	1
4	$8(6t^2-6t+1)$	$1/2 + \sqrt{3}/6$	$1/2$
6	$48(2t-1)(10t^2-10t+1)$	$1/2$ $1/2 + \sqrt{15}/10$	$(20\sqrt{10}-32)/93$ $(125-20\sqrt{10})/186$
8	$384(70t^4-140t^3+90t^2-20t+1)$	$1/2 + \sqrt{525-70\sqrt{30}}/70$ $1/2 + \sqrt{525+70\sqrt{30}}/70$	$1/4 + (41\sqrt{30}-49\sqrt{21})/12$ $1/4 - (41\sqrt{30}-49\sqrt{21})/12$

Before stating our results, we prepare further notations (see also Section II-D). Let $\delta \geq 0$ be a bound of approximation error of users' bitwise scores, and let U_i be an approximated value of $\sigma(p_i)$ for $0 \leq i \leq k$; namely $|U_i - \sigma(p_i)| \leq \delta$. Let \mathcal{R} be a positive value such that

$$\mathcal{R} \leq \min_{1 \leq \ell \leq c} \mathcal{R}_{\ell, \mathcal{P}} , \quad (5)$$

and let $\psi_1, \psi_2 > 0$ be approximated values of $\sigma(p_0)$ such that

$$\psi_1 \leq \sigma(p_0) \leq \psi_2 . \quad (6)$$

Let $\eta_1, \eta_2 > 0$ be two positive parameters, and let $x_1, x_2 > 0$ be two positive values such that

$$x_i r(x_i) \leq \eta_i \mathcal{R} \psi_1 / c , \text{ for } i = 1, 2 . \quad (7)$$

Note that $tr(t)$ is an increasing positive function for $t > 0$. Note also that the code length given by our formula below will be reduced as the inequalities (5), (6) and (7) are getting stricter. Moreover, choose values A_1 and A_2 so that, for $i = 1, 2$,

$$A_i \geq \frac{c}{(1 - \eta_1 - \eta_2/c)\mathcal{R} - 2\delta c} \cdot \frac{\psi_2}{x_i} \log \frac{1}{\varepsilon_i} , \quad (8)$$

where ε_1 and ε_2 are given security parameters related to the tracing error probability. Note also that the code length will be decreased as the inequality (8) becomes stricter.

Now we define an "approximated version" of the tracing algorithm by the following modification:

Definition 3 (approximated tracing algorithm): We modify the tracing algorithm given in Section II-B as follows. First, the approximated score \hat{S}_i of i -th user u_i is calculated by $\hat{S}_i = \sum_{j=1}^m \hat{S}_i^{(j)}$, where

$$\hat{S}_i^{(j)} = \begin{cases} U_{\nu_j} & \text{if } (y_j, w_{i,j}) = (1, 1) , \\ -U_{k-\nu_j} & \text{if } (y_j, w_{i,j}) = (1, 0) , \\ 0 & \text{if } y_j \in \{0, ?\} , \end{cases}$$

with the index ν_j defined by $p^{(j)} = p_{\nu_j}$. (Note that $|\hat{S}_i - S_i| \leq m\delta$ where S_i denotes the *true* score of u_i .) Then our approximated algorithm outputs all users whose approximated score satisfies that $\hat{S}_i \geq Z$.

Remark 4: Note that the original tracing algorithm is recovered when we take $U_\nu = \sigma(p_\nu)$ for every ν .

Now sufficient code lengths and corresponding thresholds with respect to the approximated tracing algorithm are determined by the following theorem, which will be proved in Appendix D:

Theorem 4: Choose the code length m and the threshold Z by

$$m = A_1 + A_2 , \quad (9)$$

$$Z = \left(\left(1 - \frac{\eta_2}{c} \right) \frac{\mathcal{R}}{c} - \delta \right) A_1 + \left(\frac{\eta_1 \mathcal{R}}{c} + \delta \right) A_2 \quad (10)$$

(see above for choices of the auxiliary values), and let N denote the total number of users. Then for the approximated tracing algorithm given in Definition 3, the false-positive probability is less than $1 - (1 - \varepsilon_1)^{N-1} (\leq (N-1)\varepsilon_1)$; and the false-negative probability is less than ε_2 . Hence the total tracing error probability is bounded by $(N-1)\varepsilon_1 + \varepsilon_2$, which becomes ε if we set $\varepsilon_1 = \varepsilon_2 = \varepsilon/N$.

Remark 5: Even if the value $R_{\ell, x}$ or $\mathcal{R}_{\ell, \mathcal{P}}$ is not explicitly representable on a computer's numeric system, all values \mathcal{R} , ψ_i and x_i can be chosen as being explicitly representable. Moreover, A_1 and A_2 , therefore the resulting code length, can be chosen from integers.

Here we propose the following choice of parameters

$$(\eta_1, \eta_2) = (1/2, \sqrt{c}/2) \quad (11)$$

to reduce the code length. On the other hand, the original formula in [6] can be recovered by putting $\delta = 0$, $\eta_1 = 1/4$ and $\eta_2 = c/2$ and by letting all of (5), (6), (7) and (8) be equalities.

Remark 6: Although it is somewhat complicated to compute the explicit GL distribution for large c , we can determine values \mathcal{R} , ψ_1 and ψ_2 in (5) and (6) by using inequalities (12), (31) and (32) which will be given in Section V-B and Appendix G; thus we are still able to derive some upper bounds for the code lengths even in such cases. Namely, if we put $\delta = 0$, then a sufficient code length m making the code c -secure is calculated from the above values \mathcal{R} , ψ_1 and ψ_2 as $m = A'_1 c^2 \log(1/\varepsilon_1) + A'_2 c^2 \log(1/\varepsilon_2)$, where

$$A'_i = \frac{\pi}{(1 - \eta_1 - \eta_2/c)(c+1)x_i \tan(j_1/\sqrt{(c+1)^2 + a'_2})}$$

for $i = 1, 2$ (see Section V-B and Appendix G for definitions of j_1 and a'_2). By choosing security parameters $\varepsilon_1 = \varepsilon_2 = \varepsilon/N$ as in the last statement of Theorem 4, the percentage of our code length m relative to the length $100c^2 \lceil \log(N/\varepsilon) \rceil$ of Tardos code is bounded by $A'_1 + A'_2$. Figure 1 above is thus obtained by plotting the values $A'_1 + A'_2$, where the lower and the upper curves correspond, respectively, to our choice (11) of parameters and the parameters $(\eta_1, \eta_2) = (1/4, c/2)$ recovering the code length formula given in [6].

B. Asymptotic Behavior of Code Lengths

In this subsection, we investigate properties of the GL distributions $\mathcal{P} = \mathcal{P}_c$ with c even and the asymptotic behavior of the corresponding code length determined by our formula in the limit $c \rightarrow \infty$. Proofs of results which are omitted here will be demonstrated in the Appendices.

First, we show the following bound and asymptotic behavior for the values $\mathcal{R}_{\ell, \mathcal{P}_c}$, whose proofs will be given in Appendix F:

Proposition 6: We have

$$\mathcal{R}_{\ell, \mathcal{P}_c} \geq \frac{c+1}{c\pi} \text{ for all } 1 \leq \ell \leq c, \quad (12)$$

and $\lim_{c \rightarrow \infty} \mathcal{R}_{\ell, \mathcal{P}_c} = 1/\pi$ for all $\ell \geq 1$.

Secondly, we are also able to show an asymptotic behavior of the value $\sigma(p_0)$. Here we define j_1 to be the smallest positive zero of the 0th-order Bessel function $J_0(t) = \sum_{i=0}^{\infty} (-1)^i (t/2)^{2i} / (i!)^2$ of the first kind; it is known that $j_1 = 2.40482 \dots$. Now the asymptotic behavior of $\sigma(p_0)$ is given as follows, which will be proved in Appendix G:

Proposition 7: We have $\lim_{c \rightarrow \infty} \sigma(p_0)/c = 1/j_1$.

From now, we investigate the asymptotic behavior of the code length corresponding to \mathcal{P}_c . Here, for simplicity, we put $\delta = 0$, let (5), (6), (7) and (8) be equalities, and choose parameters η_1 and η_2 so that $\lim_{c \rightarrow \infty} \eta_1 = \eta$ with $0 < \eta < \infty$, $\lim_{c \rightarrow \infty} \eta_2 = \infty$ and $\lim_{c \rightarrow \infty} \eta_2/c = \eta'$ with $0 \leq \eta' < \infty$. Moreover, we assume for a technical reason that $\log(1/\varepsilon_2)/\log(1/\varepsilon_1)$ does not diverge to ∞ when $c \rightarrow \infty$. Then we have the following result, which will be proved in Appendix H:

Theorem 5: Under the above assumptions, the code length given by (9) is asymptotically

$$m \sim \frac{\pi}{(1 - \eta - \eta')j_1 x_{\infty}} c^2 \log \frac{1}{\varepsilon_1} \text{ when } c \rightarrow \infty,$$

where x_{∞} is the unique positive value determined by $x_{\infty} r(x_{\infty}) = \eta/(\pi j_1)$.

By applying this theorem to our proposal (11) of the parameters η_1 and η_2 , we obtain the following result:

Theorem 6: Put $(\eta_1, \eta_2) = (1/2, \sqrt{c}/2)$ and assume that $\log(1/\varepsilon_2)/\log(1/\varepsilon_1)$ does not diverge to ∞ when $c \rightarrow \infty$. Then our code length is less than 20.6021% of that of Tardos code for any sufficiently large c .

Proof: In this case, we have $\eta = 1/2$ and $\eta' = 0$. By using the relations $3.14159 < \pi < 3.14160$ and $2.40482 < j_1 < 2.40483$, we have

$$x_{\infty} r(x_{\infty}) = (2\pi j_1)^{-1} > 0.06618,$$

therefore it follows that $x_{\infty} > 0.12682$ (recall that $tr(t)$ is an increasing function on $t > 0$). By these data and Theorem 5, the percentage of our code length relative to Tardos code is asymptotically

$$\frac{\pi}{(1 - \eta - \eta')j_1 x_{\infty}} < \frac{3.14160}{(1/2) \cdot 2.40482 \cdot 0.12682} < 20.6021.$$

Thus the percentage is less than 20.6021% for any sufficiently large c . ■

A similar argument can be used to show that the asymptotic percentage is slightly less than 80.7028% when we use

the parameters $(\eta_1, \eta_2) = (1/4, c/2)$ corresponding to the formula in [6]; in this case, we have $\eta = 1/4$, $\eta' = 1/2$, $x_{\infty} r(x_{\infty}) = (4\pi j_1)^{-1} > 0.03309$ and $x_{\infty} > 0.06475$. Thus the asymptotic behavior of our code is much better not only compared to Tardos code but also to [6].

C. Numerical Examples

Here we give some numerical examples of our code lengths and related parameters. We use the bias distributions given in the first part of Table III, which approximate the GL distributions, with $c \in \{2, 4, 6, 8\}$. We choose approximated bitwise scores U_i as in the second part of Table III, with approximation error $\delta = 0$ if $c = 2$ and $\delta = 10^{-5}$ if $c \in \{4, 6, 8\}$. Then Table IV gives corresponding values of \mathcal{R} , ψ_1 , ψ_2 , x_1 , x_2 , A_1 and A_2 , where we put $\varepsilon_1 = \varepsilon_2 = \varepsilon/N$, $N = 100c$, $\varepsilon = 10^{-11}$, $\eta_1 = 1/2$ and $\eta_2 = \sqrt{c}/2$. Now by (9) and (10), we obtain the resulting code lengths m and thresholds Z as in Table IV, where the row ‘%’ shows percentages of our code lengths relative to Tardos codes. On the other hand, based on results in Section V-B, further comparison of our code lengths with those of Tardos code is given by Table V, where we put $\varepsilon_1 = \varepsilon_2 = \varepsilon/N$, $N = 10^9$ and $\varepsilon = 10^{-6}$.

TABLE III
BIAS DISTRIBUTIONS \mathcal{P} AND APPROXIMATED SCORES

c	p	q	c	p	q
2	0.500 00	1.000 00	8	0.069 43	0.248 33
4	0.211 32	0.500 00		0.330 01	0.251 67
	0.788 68	0.500 00		0.669 99	0.251 67
6	0.112 70	0.332 01		0.930 57	0.248 33
	0.500 00	0.335 98			
	0.887 30	0.332 01			

c	U_0	U_1	U_2	U_3
2	1			
4	1.931 87	0.517 63		
6	2.805 90	1	0.356 39	
8	3.661 01	1.424 85	0.701 82	0.273 14

These examples show that our result in this paper indeed reduces the code lengths.

VI. REMARKS ON RECENT RELATED WORKS

At the time when the preliminary version of this paper was written, our code lengths given in Section V were to our best knowledge the shortest among known c -secure codes (at least for $c \geq 4$). After that, some recent works [7], [8], [9], [10] on Tardos code have succeeded to reduce the code lengths, by strictly improving the evaluation of tracing error probabilities and slightly modifying some parameters or even the tracing algorithm itself; their new code lengths are in fact shorter than ours. However, in their works, the problems such as large memory amount and impossibility of explicit implementation, mentioned and solved in this paper, are not concerned. For instance, their schemes still use continuous bias distributions but they did not show suitable ways to implement or approximate their continuous distributions for practical use.

Therefore, these recent results do not completely supersede the work in this paper, in particular, the most significant part

TABLE IV
AUXILIARY VALUES, LENGTHS AND THRESHOLDS FOR THE EXAMPLE

c	2	4	6	8
N	200	400	600	800
Tardos	n	12 400	51 200	115 200
	\mathcal{R}	0.5	0.408	0.377
	ψ_1	1	1.931	2.805
	ψ_2	1	1.932	2.806
	x_1	0.231	0.184	0.166
	x_2	0.315	0.347	0.377
	A_1	3622	12 907	28 878
	A_2	2656	6843	12 716
Ours	n	6278	19 750	41 594
	%	50.6	38.6	36.1
	Z	917.3...	1336.317 86	1843.450 24...
				2375.914 48...

TABLE V
ANOTHER COMPARISON OF CODE LENGTHS FOR $N = 10^9$ AND $\varepsilon = 10^{-6}$

c	4	8	16	32	64	$\rightarrow \infty$
Tardos	5.60×10^4	2.24×10^5	8.96×10^5	3.58×10^6	1.43×10^7	100%
Ours	2.18×10^4	7.72×10^4	2.78×10^5	1.01×10^6	3.75×10^6	20.6%

regarding reduction of extra memory amount. In fact, these results show that there remains a room for reducing the length of our code. Indeed, we would like to announce that our recent successive study has achieved code lengths even shorter than the abovementioned works, by using (approximation of) the GL distributions, improving our tracing algorithm, and tightly evaluating its tracing error probability. The details of the successive result will be presented in a forthcoming paper.

VII. CONCLUSION

We have discussed the problems of Tardos's fingerprinting code [12] regarding its practical use, such as large required memory and impossibility of explicit implementation, mainly due to continuity of probability distributions used in its codeword generation. We investigated the finite probability distributions used in the preceding improvement [6] of Tardos code, and determined the optimal distributions for the purpose of reducing memory amount. As Table I shows, the memory amount is indeed reduced dramatically by our result. We also reduced the code lengths significantly by improving the formula of code lengths given in [6]; and evaluated effects of approximation on security performance of our codes in a practical setting.

ACKNOWLEDGMENT

The authors would like to express their gratitude to Kazuto Ogawa and Satoshi Fujitsu at Japan Broadcasting Corporation (NHK), and to Takashi Kitagawa, Rui Zhang and Kirill Morozov at National Institute of Advanced Industrial Science and Technology (AIST), for several significant comments.

REFERENCES

- [1] D. Boneh and J. Shaw, "Collusion-secure fingerprinting for digital data," *IEEE Trans. Inform. Th.*, vol. 44-5, pp. 1897–1905, Sept. 1998.
- [2] N. L. Carothers, *Real Analysis*. Cambridge University Press, 2000.
- [3] H. Engels, *Numerical Quadrature and Cubature*. Academic Press: London, 1980.

- [4] K. J. Förster and K. Petras, "On estimates for the weights in Gaussian quadrature in the ultraspherical case," *Math. Comput.*, vol. 55, pp. 243–264, July 1990.
- [5] L. Gatteschi, "Una nuova disuguaglianza per gli zeri dei polinomi di Jacobi," *Atti Accad. Sci. Torino, Cl. Sci. Fis. Mat. Nat.*, vol. 103, pp. 259–265, 1968.
- [6] M. Hagiwara, G. Hanaoka and H. Imai, "A short random fingerprinting code against a small number of pirates," in *Proc. AAECC*, 2006, LNCS 3857, pp. 193–202.
- [7] T. Isogai and H. Muratani, "Reevaluation of Tardos's code," in *IEICE Technical Report, ISEC2006-96*, 2006, pp. 7–12.
- [8] —, "An evaluation of Tardos's asymptotic code length," presented at the 2007 Symposium on Crypto. and Inform. Security (SCIS2007), Sasebo, Japan, 2007.
- [9] B. Škorić, S. Katzenbeisser and M. U. Celik. (2007 Feb.). Symmetric Tardos fingerprinting codes for arbitrary alphabet sizes. [Online]. Available: <http://eprint.iacr.org/2007/041>
- [10] B. Škorić, T. U. Vladimirova, M. U. Celik and J. C. Talstra. (2006, July). Tardos fingerprinting is better than we thought. [Online]. Available: <http://www.arxiv.org/abs/cs/0607131>
- [11] G. Szegő, *Orthogonal Polynomials*. American Mathematical Society, 1939.
- [12] G. Tardos, "Optimal probabilistic fingerprint codes," *J. ACM*, to be published. Preliminary version: in *Proc. STOC*, 2003, pp. 116–125.

APPENDIX A PROOFS OF RESULTS IN SECTION IV-A

Here we give the proofs of our results in Section IV-A.

Proof of Proposition 2: First, the following property is easily derived from (2): $E_p[f_{\ell, \ell-x}(p)] \geq 0$ for all $2 \leq \ell \leq c$ and $1 \leq x \leq \ell - 1$ if and only if $E_p[f_{\ell, x}(p)] \leq 0$ for all $2 \leq \ell \leq c$ and $1 \leq x \leq \ell - 1$. Thus the first claim follows from Remark 1. The other claims are also straightforward by (2). ■

Proof of Lemma 1: First, an elementary analysis shows that

$$g_{\ell, x}(p) = \frac{d}{dp} (p^x(1-p)^{\ell-x}) \quad , \quad (13)$$

therefore the second claim follows from the equality

$$p^x(1-p)^{\ell-x} + p^{x+1}(1-p)^{\ell-x-1} = p^x(1-p)^{\ell-1-x} \quad .$$

Now the first claim is also derived from the relation $f_{\ell, x}(p) = g_{\ell, x}(p)\sqrt{p(1-p)}$. ■

Proof of Proposition 3: By the assumption on \mathcal{P} , we have $\mathcal{R}_{\ell,\mathcal{P}} = E_p[-f_{\ell,0}(p)]$ for any $1 \leq \ell \leq c$. Thus Lemma 1 infers that $\mathcal{R}_{\ell,\mathcal{P}} - \mathcal{R}_{\ell-1,\mathcal{P}} = E_p[f_{\ell,1}(p)]$ for any $2 \leq \ell \leq c$, therefore we have $\mathcal{R}_{\ell,\mathcal{P}} - \mathcal{R}_{\ell-1,\mathcal{P}} = 0$ by Proposition 2. Hence $\mathcal{R}_{\ell,\mathcal{P}} = \mathcal{R}_{1,\mathcal{P}} = E_p[\sqrt{p(1-p)}]$, as desired. ■

Proof of Lemma 2: We have $E_p[f_{\ell'-1,x'}(p)] = E_p[f_{\ell',x'}(p)] + E_p[f_{\ell',x'+1}(p)]$ by Lemma 1; thus all of the three terms become zero whenever any two of them are. ■

Proof of Proposition 4: Since \mathcal{P} is c -ind, Proposition 2 infers that (3) is satisfied for all parameters of the form (ℓ, x) with $2 \leq \ell \leq c$ and for $(c+1, x_0)$, where $x_0 = (c+1)/2$. Thus by Lemma 2 and induction on ν , it follows for all ν that (3) is satisfied for parameters of the form $(c+1, x_0 \pm \nu)$. Hence \mathcal{P} is $(c+1)$ -ind by Claim 1 of Proposition 2. ■

Proof of Proposition 5: By Proposition 4, it suffices for both of the two claims to prove that \mathcal{P} is c' -ind.

First, we argue the claim 1. By the assumption and Claim 3 of Proposition 2, the condition (3) is satisfied for all parameters (c', x) with $1 \leq x \leq c' - 1$, therefore Lemma 2 infers that it is also satisfied for all parameters $(c' - 1, x)$ with $1 \leq x \leq c' - 2$. Similarly, it is inductively derived that (3) is satisfied for all parameters (ℓ, x) with $2 \leq \ell \leq c'$ and $1 \leq \ell \leq \ell - 1$. Thus \mathcal{P} is c' -ind by Claim 1 of Proposition 2.

Secondly, we prove the claim 2. The assumption and Proposition 4 infer that \mathcal{P} is $(c' - 1)$ -ind, thus (3) is satisfied for all parameters $(c' - 1, x)$ with $1 \leq x \leq c' - 2$. Since (3) is satisfied for the parameter (c', x_0) in the statement, the same argument as Proposition 4 shows that (3) is also satisfied for all parameters (c', x) with $1 \leq x \leq c' - 1$. Hence \mathcal{P} is c' -ind by Claim 1 of Proposition 2. ■

APPENDIX B PROOF OF THEOREM 1

Here we give the proof of Theorem 1. First, we show that $\mathcal{P} = \mathcal{P}(\mathcal{Q})$ is a c -ind bias distribution for any symmetric QS \mathcal{Q} of degree $c - 1$. A straightforward calculation can show that this \mathcal{P} is indeed a finite probability distribution; the outputs of \mathcal{P} lie in the interval $(0, 1)$ since X is a subset of the interval $(-1, 1)$; and \mathcal{P} is symmetric since \mathcal{Q} is symmetric. Thus the remaining task is, by Claim 2 of Proposition 5, to show that $E_p[f_{\ell,1}(p)] = 0$ for all $2 \leq \ell \leq c$. Now recall the relation $f_{\ell,1}(p) = \sqrt{p(1-p)}g_{\ell,1}(p)$. Since $g_{\ell,1}$ is a polynomial of degree $\ell - 1$ ($\leq c - 1$), we have

$$\begin{aligned} & E_p[\sqrt{p(1-p)}g_{\ell,1}(p)] \\ &= \sum_{\xi \in X} \sqrt{\frac{1+\xi}{2} \cdot \frac{1-\xi}{2}} g_{\ell,1}\left(\frac{1+\xi}{2}\right) \cdot \frac{\omega(\xi)}{C\sqrt{1-\xi^2}} \\ &= \frac{1}{2C} \sum_{\xi \in X} \omega(\xi) g_{\ell,1}\left(\frac{1+\xi}{2}\right) \\ &= \frac{1}{2C} \int_{-1}^1 g_{\ell,1}\left(\frac{1+t}{2}\right) dt \\ &= \frac{1}{C} \int_0^1 g_{\ell,1}(z) dz \\ &= 0 \end{aligned} \tag{14}$$

(here (14) follows since \mathcal{Q} is a QS of degree $c - 1$, while (15) is derived from (13)). Thus $\mathcal{P}(\mathcal{Q})$ is c -ind.

Secondly, we show that $\mathcal{Q} = \mathcal{Q}(\mathcal{P})$ is a symmetric QS of degree $c - 1$ for any c -ind distribution \mathcal{P} . The set X is included in the interval $(-1, 1)$ since $0 < p_i < 1$ for all i , while \mathcal{Q} is symmetric since \mathcal{P} is symmetric. Thus the remaining task is to show that $\int_{-1}^1 F(t)dt = \sum_{\xi \in X} \omega(\xi)F(\xi)$ for any polynomial $F(t)$ of degree less than or equal to $c - 1$. Now observe that any such $F(t)$ can be expressed as a linear combination of the polynomials $g_{\ell,1}(\frac{1+t}{2})$ of degree $\ell - 1$ for $2 \leq \ell \leq c$ and a constant polynomial 1, while $\sum_i \omega(\xi_i) = 2 = \int_{-1}^1 1 dt$ by definition. Thus it suffices to show the above claim only for $F(t) = g_{\ell,1}(\frac{1+t}{2})$ with $2 \leq \ell \leq c$. For this claim, we have

$$C' \int_{-1}^1 g_{\ell,1}\left(\frac{1+t}{2}\right) dt = 2C' \int_0^1 g_{\ell,1}(z) dz = 0 \tag{16}$$

$$\begin{aligned} &= E_p[\sqrt{p(1-p)}g_{\ell,1}(p)] \\ &= C' \sum_{i=1}^k \omega(\xi_i) g_{\ell,1}\left(\frac{1+\xi_i}{2}\right) \end{aligned} \tag{17}$$

(here (16) is derived from (13), while (17) follows from Claim 1 of Proposition 2). Thus $\mathcal{Q}(\mathcal{P})$ is a symmetric QS of degree $c - 1$.

Finally, since $\sum_{\xi \in X} \omega(\xi) = 2$ and $\sum_i q_i = 1$, a straightforward computation can verify the relations $\mathcal{Q}(\mathcal{P}(\mathcal{Q})) = \mathcal{Q}$ and $\mathcal{P}(\mathcal{Q}(\mathcal{P})) = \mathcal{P}$. Hence the proof of Theorem 1 is concluded.

APPENDIX C PROOF OF THEOREM 3

Here we give the proof of Theorem 3. Put

$$\hat{L}_\nu(t) = L_\nu(2t - 1),$$

which is proportional to \tilde{L}_ν . First, note that $\hat{L}_\nu(\frac{1+\xi}{2}) = 0$ if and only if $L_\nu(\xi) = 0$, thus the set of outputs of $\mathcal{P}_c = \mathcal{P}(\mathcal{Q}_\nu)$ with $c = 2\nu$ is (by definition) the set of zeroes of \hat{L}_ν , which coincides with the set of zeroes of \tilde{L}_ν and consists of ν elements (see Theorem 2). Now note that $1 - \xi^2 = 4p(1-p)$ if $p = (1 + \xi)/2$, while

$$\begin{aligned} \frac{d}{dt} L_\nu(t) \Big|_{t=2p-1} &= \frac{d}{dt} \hat{L}_\nu\left(\frac{1+t}{2}\right) \Big|_{t=2p-1} \\ &= \frac{1}{2} \left(\frac{d}{du} \hat{L}_\nu(u) \Big|_{u=(1+t)/2} \right) \Big|_{t=2p-1} \\ &= \frac{1}{2} \frac{d}{du} \hat{L}_\nu(u) \Big|_{u=p} = C'' \tilde{L}_\nu'(p), \end{aligned}$$

where C'' is some constant. Thus the probability of \mathcal{P}_c taking the value $p = (1 + \xi)/2$ with $\xi \in X$ is

$$\begin{aligned} \frac{\omega(\xi)}{C\sqrt{1-\xi^2}} &= \frac{2}{C(1-\xi^2)^{3/2} L_\nu'(\xi)^2} \\ &= \frac{1}{4CC''^2 (p(1-p))^{3/2} \tilde{L}_\nu'(p)^2}. \end{aligned}$$

Hence the claim follows, since the factor $1/(4CC''^2)$ above is common for all p , concluding the proof.

APPENDIX D
PROOF OF THEOREM 4

Here we give the proof of Theorem 4 by evaluating the probabilities of false-negative and of false-positive. This will be done by basically the same argument as [6] except for some slight modifications.

In what follows, let \hat{x}_1 and \hat{x}_2 be two positive parameters, and put $\alpha = \hat{x}_1/\sigma(p_0)$ and $\beta = \hat{x}_2/(c\sigma(p_0))$. Before giving our proof, we recall the following fundamental tool in probability theory which is used in our argument (as well as in [6]):

Lemma 3 (Markov's Inequality): Let Y be a finite positive random variable and $t > 0$. Then we have

$$\text{Prob}(Y > t) < \frac{E[Y]}{t} \text{ and } \text{Prob}(Y \geq t) \leq \frac{E[Y]}{t},$$

where $E[Y]$ denotes the expected value of Y .

Now we give the following proposition, which is a slight modification of [6, Lemma 1] and which concerns the false-positive probability of our code:

Proposition 8 (cf. [6, Lemma 1]): Let u_i be an innocent user. For any fixed $P = (p^{(1)}, \dots, p^{(m)})$, any fixed $y = (y_1, \dots, y_m)$ and any $t > 0$, we have

$$\text{Prob}(S_i \geq t) < e^{r(\hat{x}_1)\alpha^2 m - \alpha t}, \quad (18)$$

where the probability is taken over the codewords of u_i chosen according to the P .

Proof: The proof is almost the same as that of [6, Lemma 1], except for some differences explained below. First, [6, Lemma 1] showed an inequality similar to (18) for the probability $\text{Prob}(S_i > Z)$ under the assumption that $\alpha\sigma(p_0) < \hat{x}_1$ (note that \hat{x}_1 is simply denoted by x_1 in [6]); however, the same proof is actually able to prove the same inequality for a slightly larger probability $\text{Prob}(S_i \geq Z)$ under the weaker assumption that $\alpha\sigma(p_0) \leq \hat{x}_1$. This follows from the observation that the bound $1 + r_1\alpha^2 \leq e^{r_1\alpha^2}$ used in the original proof is indeed a *strict* inequality. Secondly, [6, Lemma 1] was originally proved only when Z is the threshold, however a careful reading of the proof can reveal that the property of Z being the threshold is not used in there; therefore that proof is still valid even if Z is just an *arbitrary* positive parameter. Now our claim follows by combining these two observations. ■

Remark 7: The proof of [6, Lemma 1] is still valid (so is that of the above proposition) even if y is an *arbitrary* codeword with $y_j \in \{0, 1, ?\}$ which need not satisfy the *Marking Assumption*, only the required property of y is that it is independent of the codeword of u_i (see Assumption 2).

On the other hand, the next proposition, which is a slight modification of [6, Lemma 2], concerns the false-negative probability:

Proposition 9 (cf. [6, Lemma 2]): Let u_1, \dots, u_ℓ be the pirates with $\ell \leq c$, and $t > 0$. Then for any fixed pirates' strategy, we have

$$E \left[e^{-\beta \sum_{i=1}^{\ell} S_i} \right] \leq e^{\beta(c\beta r(\hat{x}_2) - \mathcal{R}_{\ell, P})m}, \quad (19)$$

where the expected value is taken over all P , all codewords of pirates and all y , which are chosen according to \mathcal{P} , P and the pirates' strategy, respectively. Hence by Markov's Inequality, we have

$$\begin{aligned} \text{Prob}(S_i < t \text{ for all } i) &\leq \text{Prob} \left(\sum_{i=1}^{\ell} S_i < \ell t \right) \\ &\leq \text{Prob} \left(e^{-\beta \sum_{i=1}^{\ell} S_i} > e^{-\beta \ell t} \right) \\ &< E \left[e^{-\beta \sum_{i=1}^{\ell} S_i} \right] / e^{-\beta \ell t} \\ &\leq e^{\beta(c\beta r(\hat{x}_2) - \mathcal{R}_{\ell, P})m + \beta \ell t}. \end{aligned}$$

Proof: The proof is basically the same as [6, Lemma 2]; it also works in our situation by noticing the following points only. First, the original proof allows the pirates' strategy to be probabilistic, though it was not clarified. Secondly, although [6] only considers the restricted case that y contains no bit '?', an argument appeared in [12] can generalize the proof in [6] to our situation where y may contain '?'. ■

Now we start to prove Theorem 4. First, recall that in the approximated tracing algorithm given in Definition 3, a user is outputted if and only if $\hat{S} \geq Z$ where \hat{S} denotes the approximated score. Now $\hat{S} \geq Z$ infers $S \geq Z - m\delta$ and $\hat{S} < Z$ infers $S < Z + m\delta$ by definition of δ ; thus to achieve $\text{Prob}(u_{i_0} \text{ is outputted}) < \varepsilon_1$, where u_{i_0} is an arbitrarily fixed innocent user, and to achieve $\text{Prob}(\text{no pirate is outputted}) < \varepsilon_2$ as well, it suffices to satisfy the following two bounds

$$\text{Prob}(S \geq Z - m\delta) < \varepsilon_1, \quad (20)$$

$$\text{Prob}(S_i < Z + m\delta \text{ for all } i) < \varepsilon_2, \quad (21)$$

where S denotes an arbitrarily fixed innocent user's true score and S_1, \dots, S_ℓ (with $\ell \leq c$) denote the ℓ pirates' true scores. Now by Propositions 8 and 9, the following conditions yield (20) and (21):

$$r(\hat{x}_1)\alpha^2 m - \alpha(Z - m\delta) \leq \log \varepsilon_1,$$

$$\beta(c\beta r(\hat{x}_2) - \mathcal{R}_{\ell, P})m + \beta\ell(Z + m\delta) \leq \log \varepsilon_2.$$

Moreover, since the values m , Z , β and δ are all nonnegative, the following conditions

$$r(\hat{x}_1)\alpha^2 m - \alpha(Z - m\delta) = \log \varepsilon_1, \quad (22)$$

$$\beta(c\beta r(\hat{x}_2) - \mathcal{R})m + \beta c(Z + m\delta) = \log \varepsilon_2 \quad (23)$$

also yield (20) and (21). Now if we solve equations (22) and (23) in m and Z , where \hat{x}_1 and \hat{x}_2 are determined by $\hat{x}_i r(\hat{x}_i) = \sigma(p_0)\eta_i \mathcal{R}/c$ for $i = 1, 2$ and α and β are determined as above, then the code length m and the threshold Z are given by

$$m = \frac{c\sigma(p_0)}{C} \left(\frac{1}{\hat{x}_1} \log \frac{1}{\varepsilon_1} + \frac{1}{\hat{x}_2} \log \frac{1}{\varepsilon_2} \right) \quad (24)$$

(where $C = (1 - \eta_1 - \eta_2/c)\mathcal{R} - 2\delta c$) and

$$Z = \frac{\sigma(p_0)}{C} \left(\frac{(1 - \eta_2/c)\mathcal{R} - \delta c}{\hat{x}_1} \log \frac{1}{\varepsilon_1} + \frac{\eta_1 \mathcal{R} + \delta c}{\hat{x}_2} \log \frac{1}{\varepsilon_2} \right) \quad (25)$$

which generalize the formula in [6] (the original is recovered by putting $\eta_1 = 1/4$, $\eta_2 = c/2$ and $\delta = 0$). Moreover, if we

take the values ψ_i , x_i and A_i ($i = 1, 2$) as in Section V-A, and determine the modified code length \hat{m} and threshold \hat{Z} by

$$\hat{m} = A_1 + A_2, \quad (26)$$

$$\hat{Z} = \left(\left(1 - \frac{\eta_2}{c} \right) \frac{R}{c} - \delta \right) A_1 + \left(\frac{\eta_1 R}{c} + \delta \right) A_2, \quad (27)$$

then by comparing the pair of (24) and (25) with the pair of (26) and (27), we can show that conditions (20) and (21) are satisfied with ε_i replaced by e^{-k_i} where

$$k_i = \frac{C}{c\sigma(p_0)} x_i A_i \text{ for } i = 1, 2.$$

Since $\psi_2 \geq \sigma(p_0)$ and $x_i \leq \hat{x}_i$ ($i = 1, 2$) by definition, we have $e^{-k_i} \leq \varepsilon_i$ by the choice of A_i . Thus the code length \hat{m} and threshold \hat{Z} , which are precisely those chosen in Theorem 4 (see (9) and (10)), provide the desired security performance. Hence the proof of Theorem 4 is concluded.

APPENDIX E

A LEMMA FOR PROOF OF PROPOSITION 6

Here we prepare the following well-known fact in elementary analysis, which will be used in the proof of Proposition 6 given in Appendix F:

Lemma 4: Let $\{f_n\}_{n=1}^\infty$ be a sequence of nonnegative continuous functions on the same open interval $I = (a, b)$, whose sum $\sum_{n=1}^\infty f_n$ converges to a continuous function f at every point in I . If all of the improper integrals $\int_a^b f_n(x) dx$ and $\int_a^b f(x) dx$ exist and converge, then $\lim_{n \rightarrow \infty} \int_a^b \sum_{i=1}^n f_i(x) dx = \int_a^b f(x) dx$.

From now, we give a proof of this result for the sake of completeness. In the proof, we use the following two facts, which can be found in most of undergraduate textbooks of elementary analysis:

Lemma 5 (Dini's Theorem; see e.g. [2, p.151]): Let $I = [a, b]$ be a closed interval, and let $\{g_i\}_{i=1}^\infty$ be an increasing sequence of continuous functions g_i on I which converges to another continuous function g at every point in I ; i.e. $g_{i-1}(x) \leq g_i(x) \rightarrow g(x)$ when $x \in I$. Then the convergence is uniform; i.e. for any $\varepsilon > 0$, there is an index n such that $|g(x) - g_i(x)| < \varepsilon$ for every $i \geq n$ and $x \in I$.

Lemma 6 (see e.g. [2, Theorem 10.5]): Let $\{g_i\}_{i=1}^\infty$ be a sequence of continuous functions on the same closed interval $I = [a, b]$ which converges uniformly to a function g on I . Then $\int_a^b g_i(x) dx$ converges to $\int_a^b g(x) dx$ when $i \rightarrow \infty$.

Now we start to prove Lemma 4. First, note that the function f is nonnegative on I by the assumption. Then, given an arbitrary $\varepsilon > 0$, the assumption on convergence of the improper integral $\int_a^b f(x) dx$ infers that

$$0 \leq \int_a^b f(x) dx - \int_{a'}^{b'} f(x) dx < \frac{\varepsilon}{2} \quad (28)$$

for some $a < a' \leq b' < b$. Now by the assumption, the sequence $\{g_i\}_{i=1}^\infty$ defined by $g_i = \sum_{n=1}^i f_n$ is an increasing sequence of continuous functions on the closed interval $I' = [a', b']$ which converges to the continuous function f at every point in I' . Thus Lemma 5 infers that the convergence

is uniform; it follows that $\int_{a'}^{b'} g_i(x) dx \rightarrow \int_{a'}^{b'} f(x) dx$ when $i \rightarrow \infty$ by Lemma 6. Therefore there is an index n such that

$$\int_{a'}^{b'} f(x) dx - \int_{a'}^{b'} g_i(x) dx < \frac{\varepsilon}{2} \text{ for any } i \geq n. \quad (29)$$

By combining (28) and (29), we have

$$\begin{aligned} 0 &\leq \int_a^b f(x) dx - \int_a^b g_i(x) dx \\ &\leq \int_a^b f(x) dx - \int_{a'}^{b'} g_i(x) dx < \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon \end{aligned}$$

for any $i \geq n$ (note that $0 \leq g_i \leq f$ for each i). This means that $\int_a^b g_i(x) dx$ converges to $\int_a^b f(x) dx$ when $i \rightarrow \infty$, as desired. Hence the proof of Lemma 4 is concluded.

APPENDIX F

PROOF OF PROPOSITION 6

Here we give the proof of Proposition 6. First, let $\mathcal{Q}_n = (X_n, \omega_n)$ denote the Gauss-Legendre QS of order n (see Section IV-C for definition), therefore the set X_n consists of n zeroes of the Legendre polynomial $L_n(t)$. In our proof of Proposition 6, we use the following result, which is directly derived from the latter part of Inequality (2.18) in [4, Corollary 1] by choosing the parameter $\lambda = 1/2$:

Lemma 7 ([4, Corollary 1]): We have

$$\frac{\omega_n(\xi)}{\sqrt{1-\xi^2}} \leq \frac{\pi}{n+1/2}$$

for all $n \geq 1$ and all $\xi \in X_n$.

Now we start the proof of Proposition 6. Recall that c is now assumed to be even; put $c = 2n$. First, by combining Proposition 3 and Theorem 1, we have

$$\begin{aligned} \mathcal{R}_{\ell, \mathcal{P}_c} &= E_p \left[\sqrt{p(1-p)} \right] \\ &= \sum_{\xi \in X_n} \sqrt{\frac{1+\xi}{2} \cdot \frac{1-\xi}{2}} \cdot \frac{\omega_n(\xi)}{C_n \sqrt{1-\xi^2}} \\ &= \frac{1}{2C_n} \sum_{\xi \in X_n} \omega_n(\xi) = \frac{1}{C_n}, \end{aligned}$$

where $C_n = \sum_{\xi \in X_n} \omega_n(\xi) / \sqrt{1-\xi^2}$. Now Lemma 7 gives us that $C_n \leq n\pi / (n+1/2) = c\pi / (c+1)$, thus the former claim follows.

For the proof of the latter claim, it suffices to show that $\lim_{n \rightarrow \infty} C_n = \pi$. Now it follows that

$$\frac{1}{\sqrt{1-t^2}} = \sum_{i=0}^{\infty} \frac{1}{4^i} \binom{2i}{i} t^{2i} \geq \sum_{i=0}^{n-1} \frac{1}{4^i} \binom{2i}{i} t^{2i}$$

for $-1 < t < 1$, therefore we have

$$\begin{aligned} \frac{n\pi}{n+1/2} &\geq C_n \geq \sum_{\xi \in X_n} \omega_n(\xi) \sum_{i=0}^{n-1} \frac{1}{4^i} \binom{2i}{i} \xi^{2i} \\ &= \int_{-1}^1 \sum_{i=0}^{n-1} \frac{1}{4^i} \binom{2i}{i} t^{2i} dt \end{aligned} \quad (30)$$

(here (30) follows from the fact that the QS \mathcal{Q}_n is of degree $2n-1$). Now owing to Lemma 4 in Appendix E with

$I = (-1, 1)$ and $f_i(t) = 4^{-i} \binom{2i}{i} t^{2i}$, the value (30) converges to $\int_{-1}^1 \sum_i f_i(t) dt = \int_{-1}^1 (1 - t^2)^{-1/2} dt = \pi$ when $n \rightarrow \infty$, while $n\pi/(n + 1/2)$ also converges to π . Thus we have $\lim_{n \rightarrow \infty} C_n = \pi$, as desired. Hence the proof of Proposition 6 is concluded.

APPENDIX G PROOF OF PROPOSITION 7

Here we give the proof of Proposition 7. First, we define θ_n to be the unique value such that $0 < \theta_n < \pi$ and $-\cos \theta_n$ is the smallest zero of the Legendre polynomial $L_n(t)$ (recall that the zeroes of $L_n(t)$ lie in the open interval $(-1, 1)$). Then we have the following result, which will be used in the proof of Proposition 7:

Lemma 8 ([5, p.264]): For any n , we have

$$\frac{j_1}{\sqrt{(n+1/2)^2 + a_2}} < \theta_n < \frac{j_1}{\sqrt{(n+1/2)^2 + a_1}},$$

where $a_1 = 1/12$ and $a_2 = 1/4 - 1/\pi^2$.

From now, we prove Proposition 7. First, by the definitions of p_0 and θ_n , Theorem 1 infers that

$$p_0 = \frac{1 - \cos \theta_n}{2} = \sin^2 \frac{\theta_n}{2},$$

therefore $\sigma(p_0) = 1/\tan(\theta_n/2)$. Thus by Lemma 8, we have

$$\frac{1}{\tan(j_1/\sqrt{(c+1)^2 + a_1'})} < \sigma(p_0) \quad (31)$$

$$< \frac{1}{\tan(j_1/\sqrt{(c+1)^2 + a_2'})}, \quad (32)$$

where $a_i' = 4a_i$ (namely $a_1' = 1/3$ and $a_2' = 1 - 4\pi^{-2}$). Moreover, an elementary analysis gives us

$$\begin{aligned} & \frac{1}{c \tan(j_1/\sqrt{(c+1)^2 + a_i'})} \\ &= \frac{\cos(j_1/\sqrt{(c+1)^2 + a_i'})}{j_1/\sqrt{(1+1/c)^2 + a_i'/c^2}} \cdot \frac{j_1/\sqrt{(c+1)^2 + a_i'}}{\sin(j_1/\sqrt{(c+1)^2 + a_i'})} \end{aligned}$$

for $i = 1, 2$, which converges to $1/j_1$ when $c \rightarrow \infty$. Hence the claim of Proposition 7 follows, concluding the proof.

APPENDIX H PROOF OF THEOREM 5

Here we give the proof of Theorem 5. First, we have $\mathcal{R} = \mathcal{R}_{\ell, \mathcal{P}} \rightarrow 1/\pi$ by Proposition 6 (see also Proposition 3) and $c^{-1}\sigma(p_0) \rightarrow 1/j_1$ by Proposition 7 when $c \rightarrow \infty$. Thus the parameter x_1 , which is determined by $x_1 r(x_1) = \eta_1 \mathcal{R} \sigma(p_0)/c$, converges when $c \rightarrow \infty$ to x_∞ given in the statement (note that the continuous function $tr(t)$ is strictly increasing for $t > 0$ and its image is the whole infinite interval $(0, +\infty)$, therefore such x_∞ is uniquely determined). Similarly, the other parameter x_2 converges to ∞ when $c \rightarrow \infty$, since $x_2 r(x_2) = \eta_2 \mathcal{R} \sigma(p_0)/c \rightarrow \infty$. Now by (9), we have

$$\begin{aligned} & \frac{m}{c^2 \log(1/\varepsilon_1)} \\ &= \frac{1}{\mathcal{R}(1 - \eta_1 - \eta_2 c^{-1})} \cdot \frac{\sigma(p_0)}{c} \left(\frac{1}{x_1} + \frac{\log(1/\varepsilon_2)}{x_2 \log(1/\varepsilon_1)} \right), \end{aligned}$$

while the above argument shows that

$$\mathcal{R}(1 - \eta_1 - \eta_2 c^{-1}) \rightarrow \frac{1 - \eta - \eta'}{\pi} \text{ and } \frac{\sigma(p_0)}{c} \rightarrow \frac{1}{j_1}$$

when $c \rightarrow \infty$. Moreover, by the assumption on value $\log(1/\varepsilon_2)/\log(1/\varepsilon_1)$ we have that

$$\frac{\log(1/\varepsilon_2)}{x_2 \log(1/\varepsilon_1)} \rightarrow 0 \text{ when } c \rightarrow \infty.$$

Thus we have

$$\frac{m}{c^2 \log(1/\varepsilon_1)} \rightarrow \frac{\pi}{(1 - \eta - \eta') j_1 x_\infty},$$

therefore the claim follows. Hence the proof of Theorem 5 is concluded.